WHAT IS CLAIMED IS:

1. A software tool comprising:

portability code to include an intermediate representation of source code with an executable representation of the source code, wherein the intermediate representation of the source code includes information sufficient for generating another executable representation of the source code.

2. The software tool of claim 1, further comprising the portability code to include source code processing information with the intermediate source code representation and the executable source code representation.

3. The software tool of claim 2, wherein the source code processing information includes one or more of compiler directives, compiler options, compiler flags, virtual machine options, and source code processing invocation commands.

4. The software tool of claim 1, wherein the executable source code representation at least includes executable code.

5. The software tool of claim 1, wherein the intermediate representation includes one or more of linking information, symbol tables, object bindings, and platform independent optimization information.

6. The software tool of claim 1, wherein the portability code allots sufficient space for the executable representation and for the intermediate representation as a portable executable source code representation.

7. The software tool of claim 1, further comprising platform independent processing code to perform platform independent processing of the source code and to generate the intermediate representation of the source code.

8. The software tool of claim 7, wherein the platform independent processing includes one or more of lexical analysis, syntax analysis, platform independent optimization, and semantic analysis.

9. The software tool of claim 1, wherein the software tool uses the intermediate source code representation to generate a second executable representation of the source code.

10. The software tool of claim 9, wherein the second executable source code representation is for a different one or more of platform and operating environment than the executable source code representation.

11. The software tool of claim 9, further comprising preservation code to prohibit modifying optimizations of the intermediate representation.

12. The software tool of claim 1, wherein the software tool includes one or more of a compiler front-end, a compiler back-end, an interprocedural optimizer, an interpreter, and a linker.

13. The software tool of claim 1, further comprising the portability code to include linking information with the executable source code representation and the intermediate source code representation.

14. The software tool of claim 13, wherein the linking information includes object file information.

15. The software tool of claim 13, further comprising a link-time optimizer to re-link object files corresponding to the source code in accordance with the linking information.

16. A method comprising:

including an intermediate representation of source code with an executable representation of the source code, wherein the executable source code representation at least includes executable code and the intermediate

source code representation is generated from platform independent processing of the source code; and

indicating the intermediate source code representation relative to the executable source code representation, wherein the intermediate source code representation is providable for generation of one or more other executable representations of the source code.

17. The method of claim 16 further comprising including source code processing command information in the portable executable source code representation.

18. The method of claim 17, wherein the source code processing command information includes one or more of compiler directives, compiler flags, compiler options, and virtual machine options.

19. The method of claim 16, wherein the intermediate representation includes one or more of linking information, symbol tables, object bindings, and optimization information.

20. The method of claim 16 further comprising performing the platform independent processing of the source code to generate the intermediate source code representation.

21. The method of claim 20, wherein platform independent processing includes one or more of lexical analysis, syntax analysis, platform independent optimization, and semantic analysis.

22. The method of claim 16 further comprising generating a portable executable file with space sufficient to accommodate the executable source code representation and the intermediate source code representation.

23. The method of claim 22 further comprising:

extracting the intermediate representation from the portable executable file; and

supplying the extracted intermediate representation for generation of a second executable representation, wherein the second executable representation and the executable representation undergo different platform dependent processing.

24. The method of claim 23, wherein the platform dependent processing includes one or more of storage allocation and executable code generation.

25. The method of claim 16 further comprising including a second intermediate representation of the source code with the executable source code representation, wherein the second intermediate representation of the source code is providable with the intermediate representation of the source code for generation of the second executable representation of the source code, wherein the intermediate representation is generated from a first unit of the source code and the second intermediate representation is generated from a second unit of the source code.

26. The method of claim 16 embodied as a computer program product encoded on one or more machine-readable media.

27. A method comprising:

using an intermediate representation of a first source code included with a first executable representation of the first source code and an indication of source code processing command information that corresponds to the intermediate representation; and

generating a second executable representation of the first source code with the intermediate representation of the first source code, wherein the first and second executable representations are respectively for a first and a second platforms.

28. The method of claim 27 further comprising:

linking a second source code's representation with the second executable representation.

29. The method of claim 28 wherein the second source code's representation includes one or more of an intermediate representation of the second source code and an executable representation of the second source code.

30. The method of claim 28 wherein the second source code's representation includes a library.

31. The method of claim 28 further comprising:

including linking information that corresponds to the first and second source codes.

32. The method of claim 27 further comprising including the intermediate representation with the first executable representation.

33. The method of claim 27 further comprising performing optimizations on sections of the intermediate representation that are not indicated as non-modifiable.

34. The method of claim 27 embodied as a computer program product encoding on one or more machine-readable media.

35. A computer program product encoded in one or more machine-readable media, wherein the computer program product comprises:

a first sequence of instructions operable to store source code processing command that corresponds to generation of an intermediate representation of a source code; and

a second sequence of instructions operable to include the intermediate source code representation and the stored source code processing command information with an executable representation of the source code.

36. The computer program product of claim 35, wherein the executable source code representation includes executable code and data.

37. The computer program product of claim 35 further comprising a third sequence of instructions operable to determine sufficient space for the intermediate

source code representation, the source code processing command information, and the executable source code representation, and operable to supply indication of the sufficient space.

38. The computer program product of claim 35 further comprising a third sequence of instructions operable to generate a second executable representation from the intermediate representation.

39. The computer program product of claim 35 further comprising a third sequence of instructions operable to mark sections of source code to prevent optimizations inconsistent with the source code and to carry forward those markings into the intermediate representation of the source code.

40. The computer program product of claim 35 further comprising a third sequence of instructions operable to perform platform independent processing of the source code to generate the intermediate representation of the source code.

41. The computer program product of claim 40, wherein platform independent processing of the source code includes one or more of lexical analysis, semantic analysis, syntax analysis, and platform independent optimizations.

42. A computer program product encoded in one or more machine-readable media, wherein the computer program product comprises:

a first sequence of instructions operable to locate a source code's intermediate representation that is included with a first executable representation of the source code; and

a second sequence of instructions operable to utilize the intermediate representation for generation of a second executable representation of the source code.

43. The computer program product of claim 42 further comprising:

the first sequence of instructions operable to locate source code processing command information; and

the second sequence of instructions operable to utilize the source code processing command information and the intermediate representation for said generation of the second executable representation of the source code.

44. The computer program product of claim 42, wherein generation of the second executable representation comprises performing platform dependent processing.

45. The computer program product of claim 44 further comprising a third sequence of instructions operable to perform the platform dependent processing.

46. The computer program product of claim 44, wherein the platform dependent processing includes one or more of storage allocation, platform dependent optimizations, and target platform code generation.

47. The computer program product of claim 42 further comprising a third sequence of instructions operable to include a source code's intermediate representation with an executable representation of the source code.

48. The computer program product of claim 47, wherein the third sequence of instructions are further operable to include source code processing command information with a source code's intermediate representation with an executable representation of the source code.

49. An apparatus comprising:
means for translating source code; and
means for including one or more intermediate representations of source code with an executable representation of the source code.

50. The apparatus of claim 49 further comprising means for including source code processing command information with intermediate source code representations and an executable source code representation.

51. The apparatus of claim 49 further comprising means for preventing code optimizations that are inconsistent with the source code.

52. The apparatus of claim 49 wherein the intermediate representations include one or more of platform independent optimizations, symbol tables, and object bindings.

53. The apparatus of claim 49 further comprising means for utilizing one or more intermediate representations of source code that accompany a first executable representation of the source code and for generating a second executable representation of the source code with the intermediate representations.

54. An apparatus comprising:

means for extracting one or more intermediate representations of source code from a portable executable file, wherein the portable executable file includes a first executable representation of the source code along with the intermediate representations of the source code; and

means for generating a second executable representation of the source code with the extracted intermediate representation.

55. The apparatus of claim 54 further comprising means for including intermediate representations of source code with an executable representation of the source code in a portable executable file.

56. The apparatus of claim 54 wherein the first executable representation and the second executable representation are for a respective first platform and second platform.

57. A machine-readable medium encoding comprising:

a portable executable representation of source code, wherein the portable executable representation includes,

one or more intermediate representations of the source code,

an executable representation of the source code, and

source code processing command information.

58. The machine-readable medium encoding of claim 57, wherein the intermediate representations include one or more of object bindings, platform dependent optimizations, and symbol tables.

59. The machine-readable medium encoding of claim 57, wherein the executable representation of the source code includes executable code and data.

60. The machine-readable medium encoding of claim 57, wherein the source code processing command information includes one or more of compiler options, compiler directives, and compiler flags.